From bounded T-systems to 1-safe T-systemsup to language equivalence

Philippe Darondeau, Harro Wimmel

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

From bounded T-systems to 1-safe T-systems up to language equivalence

Philippe Darondeau — Harro Wimmel

N° 4708

Janvier 2003

THÈME 1

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--4708--FR+ENG

From bounded T-systems to 1-safe T-systems up to language equivalence

Philippe Darondeau * , Harro Wimmel †

Thème 1 — Réseaux et systèmes
Projet S4

Rapport de recherche n° 4708 — Janvier 2003 — 15 pages

Abstract: We show that every finite and bounded marked graph or T-system has a 1-safe labelled version with an identical language.

Key-words: Petri nets; T-systems; marked graphs; sequential components; circuits; Arnold-Nivat product; labelling; permutations.

* IRISA, campus de Beaulieu, F35042 Rennes Cedex <darondeau@irisa.fr>
† University of Oldenburg, Fachbereich Informatik, P.O.Box 25 03, D-26111 Oldenburg <Harro.Wimmel@Informatik.Uni-Oldenburg.DE>

Une transformation des T-systèmes bornés en des T-systèmes 1-saufs engendrant le même langage

Résumé : Nous montrons que tout T-système fini et borné peut être transformé en un T-système étiqueté 1-sauf qui engendre le même langage.

Mots-clés : Réseaux de Petri; graphes marqués; T-systèmes; composantes séquentielles; circuits; produit d'Arnold-Nivat; étiquetage; permutations.

1 Introduction

The goal of this paper is to answer Wolgang Reisig's request made and forwarded to the Petri Nets Mailing List in March 2001. Wolfgang Reisig wrote in [6]:

I would like to see a nice proof of the following fairly obvious problem:
Let G denote the set of all marked graphs with labelled transitions.
Each N of G defines a formal language.
Claim*: For each N of G there exists an N' in G defining the same language,*
such that N' is 1-safe.

We shall partially answer this request by giving a construction of (finite) N' for finite and bounded marked graphs N. The assumption that all marked graphs under consideration are *bounded* was not explicit in the statement of the problem. As the set of all prefixes of words in ${D'_1}^*$ (the restricted Dyck language over one pair of parentheses) may be recognized by an unbounded marked graph (with one place that counts the unmatched left parentheses), and seeing that ${D'_1}^*$ is not regular, a finite unbounded marked graph cannot always be transformed into a finite 1-safe marked graph with the same language. However, the constraint that N' should be finite was not explicit either in the statement of the problem. We are inclined to believe that the problem has still a solution in the set of infinite 1-safe marked graphs N' when N is an unbounded marked graph, as examplified in Fig. 1, but further work is needed before this conjecture can be fixed. We shall actually solve the above problem for finite and bounded T-systems, which extend smoothly over marked graphs. To be more precise, we shall solve the problem for unlabelled T-systems, as the solution for labelled T-systems follows then immediately. The rest of the paper is organized as follows. Section 2 recalls the basic definition of T-systems and marked graphs and their languages. Section 3 reduces the problem for finite and bounded T-systems to a similar problem for finite and bounded, live and strongly connected marked graphs. We show in sections 4 and 5 that such marked graphs may be simulated up to language equivalence by finite sets of cyclic processes synchronized by the Arnold-Nivat product [1]. The simulation relies heavily on Finkel-Memmi *fifo* nets [4] used as a bridge between marked graphs and communicating sequential processes. Section 6 converts the synchronized products of cyclic processes to 1-safe marked graphs, so it brings the solution to the problem that was to solve. It is also observed that the resulting 1-safe marked graphs are live. However, they need not be strongly connected. Section 7 shows that anyway all maximal strongly connected components of the resulting 1-safe marked graph are isomorphic up to initial markings. This fact is illustrated in section 8, where the constructions are put to work on a simple case study. Section 9 looks briefly at the implications of the results.

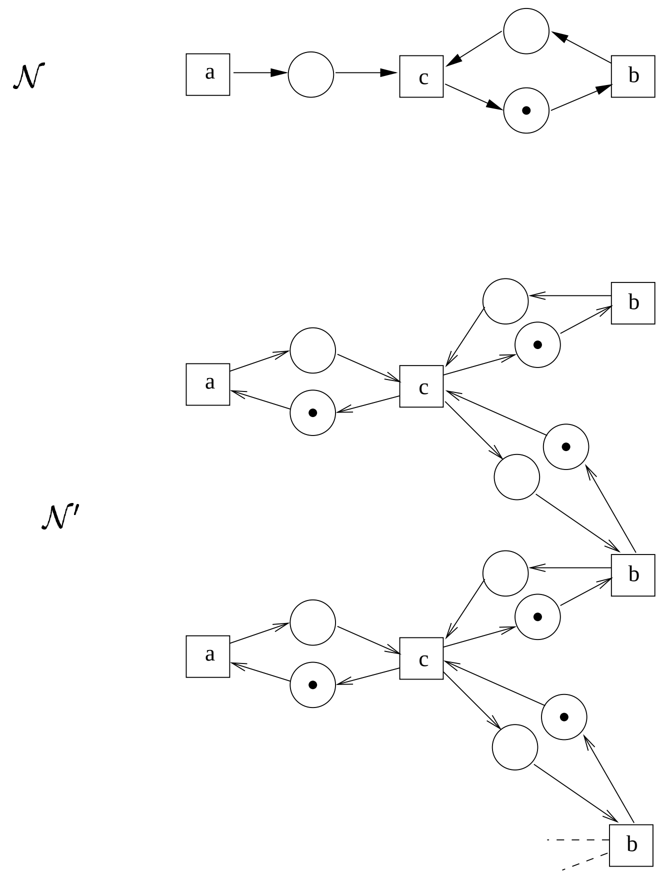

Figure 1: An unbounded marked graph and its infinite 1-safe version

2 The terms of the problem for T-systems

We recall here the basic definitions of ordinary P/T-nets, T-systems, and marked graphs before stating the problem addressed in the paper.

Definition 2.1 (finite ordinary P/T-nets) *A* finite ordinary P/T-net *is a bipartite graph $N = (P, T, F)$, where P and T are finite and disjoint sets of vertices, called* places *and* transitions*, respectively, and $F : (P \times T) \cup (T \times P) \to \{0, 1\}$ is the characteristic function of the set of directed edges of the graph. A* marking *of N is a map $m : P \to \mathbb{N}$. The* state graph *of N is a labelled multi-graph, with markings as vertices, where there is an edge labelled with transition t from m to m' (in notation: $m[t\rangle m'$) if and only if, for every place $p \in P$, $m(p) \geq F(p, t)$ and $m'(p) = m(p) - F(p, t) + F(t, p)$. The* reachable state graph *of an* initialized *P/T-net $\mathcal{N} = (P, T, F, m_0)$ with* initial marking *m_0 is the restriction of its state graph to the markings that may be reached from m_0. $\mathcal{N}$ is* initially live *if every transition $t \in T$ is the label of an edge of its reachable state graph; $\mathcal{N}$ is* live *if, for every reachable marking m of $\mathcal{N}$, the net (P, T, F, m) is initially live. A finite P/T-net is* bounded *if its reachable state graph is finite. A bounded P/T-net is* 1-safe *if all its reachable markings are maps ranging over $\{0, 1\}$ or subsets thereof. The* free language *$L(\mathcal{N})$ of a bounded P/T-net $\mathcal{N}$ is the set of all words in T^* that are accepted by its reachable state graph, considered as an automaton with initial state m_0 and with all states final. The* labelled language *$L(\mathcal{N}, \lambda)$ of a bounded P/T-net $\mathcal{N}$ with transition* labelling map *$\lambda : T \to \Sigma$ is the image of its free language under the unique monoid morphism $\lambda : T^* \to \Sigma^*$ that extends this labelling map.*

Definition 2.2 (T-systems and marked graphs) *An initialized ordinary Petri net (P, T, F, m_0) is a* T-system *if $(\forall p \in P)$ $(\,|{}^\bullet p| \leq 1 \wedge |p^\bullet| \leq 1\,)$ where ${}^\bullet p = \{t \in T \mid F(t, p) \neq 0\}$ and $p^\bullet = \{t \in T \mid F(p, t) \neq 0\}$. A T-system is a* marked graph *if $(\forall p \in P)$ $(\,|{}^\bullet p| = 1 \wedge |p^\bullet| = 1\,)$.*

Problem 2.3 *Given any finite and bounded T-system $\mathcal{N} = (P, T, F, m_0)$, is there a finite and 1-safe T-system $\mathcal{N}' = (P', T', F', m'_0)$ and a transition labelling map $\lambda : T' \to T$ such that the labelled language of $\mathcal{N}'$ is equal to the free language of $\mathcal{N}$,* i.e. *in formulas, such that $L(\mathcal{N}) = L(\mathcal{N}', \lambda)$?*

3 Reducing the problem to marked graphs

In order to ease solving problem 2.3, we show in this section how to reduce this problem on T-systems to a similar problem on strongly connected marked graphs. We proceed in two stages: problem 2.3 is reduced first from T-systems to marked graphs, next from marked graphs to strongly connected marked graphs. Let us recall that marked graphs (P, T, F, m) deserve this name since they may be considered as graphs as follows: the set of vertices is T, the set of (directed) edges is P, p is an edge from t to t' if $F(t, p) = 1$ and $F(p, t') = 1$, and furthermore this edge bears the integral mark $m(p)$. Strong connectedness has its usual meaning in this context.

Let $\mathcal{N} = (P, T, F, m_0)$ be a finite and bounded T-system. If there are *dead* transitions $t \in T$, *i.e.* transitions that cannot be fired at any reachable marking, removing from $\mathcal{N}$ all such transitions does not alter $L(\mathcal{N})$. Therefore, one may assume w.l.o.g. that every transition $t \in T$ is initially live. Let us next consider places. Suppose $|p^\bullet| = 0$ for some place $p \in P$, then removing this place does not alter the language of the T-system. Suppose $|{}^\bullet p| = 0$ for some place p such that $m_0(p) = 0$, then removing this place together with the (unique) transition t such that $F(p,t) = 1$, if it exists, does not alter the language of the T-system. By eliminating iteratively all places that meet either case sooner or later, $\mathcal{N}$ may be contracted to an equivalent T-system such that $|p^\bullet| = 1$ for every place p and $m_0(p) > 0$ whenever $|{}^\bullet p| = 0$. Suppose now $|{}^\bullet p| = 0$ for some place p, and let t denote the (unique) transition such that $F(p,t) = 1$. Define an augmented T-system $\mathcal{N}_\sharp$ with one place $p_\sharp$ added to the set P, one transition $t_\sharp$ added to the set T, and with extended maps F and m_0 such that $F(t, p_\sharp) = 1$, $F(p_\sharp, t_\sharp) = 1$, $F(t_\sharp, p) = 1$, and $m_0(p_\sharp) = 0$. As the transition t is initially live, the new transition $t_\sharp$ is also. Suppose that $L(\mathcal{N}_\sharp) = L(\mathcal{N}', \lambda)$ holds for some 1-safe T-system $\mathcal{N}'$ with the set of transitions T' and for some labelling map $\lambda : T' \to T \cup \{t_\sharp\}$. As $L(\mathcal{N}) = L(\mathcal{N}_\sharp) \cap T^*$, it follows that $L(\mathcal{N}) = L(\mathcal{N}'_\flat, \lambda_\flat)$ where $\mathcal{N}'_\flat$ is the T-system obtained from $\mathcal{N}'$ by restricting its set of transitions T' to $T' \cap \lambda^{-1}(T)$ and $\lambda_\flat$ is the restriction of λ to this subset. Hence, by augmenting simultaneously $\mathcal{N}$ with one place $p_\sharp$ for each place p such that $|{}^\bullet p| = 0$, problem 2.3 may be reduced to initially live T-systems in which $|{}^\bullet p| = 1$ and $|p^\bullet| = 1$ for every place p, that is to say, to initially live marked graphs.

Let now $\mathcal{N} = (P, T, F, m_0)$ be an initially live, finite and bounded marked graph. The assumption of boundedness entails that whenever a transition $t \in T$ is devoid of input places (*i.e.* when $F(p,t) = 0$ for all $p \in P$), this transition must be isolated. However, there may exist non-isolated transitions $t \in T$ without any output place (*i.e.* such that $F(t,p) = 0$ for all $p \in P$). For any transition t in this case, one may supplement each input place p of t with a complementary place $\overline{p}$ without altering $L(\mathcal{N})$ –where places p and $\overline{p}$ are said to be *complementary* if $F(p,t) + F(\overline{p},t) = 1$ and $F(t,p) + F(t,\overline{p}) = 1$ for all $t \in T$, and $m_0(p) + m_0(\overline{p})$ is the maximal value reached by $m(p)$ when m ranges over the reachable markings of $\mathcal{N}$. This transformation preserves the initial liveness of $\mathcal{N}$ and the respective bounds reached by the original places of $\mathcal{N}$. Moreover, after the transformation, every maximal connected component of $\mathcal{N}$ is strongly connected. In fact, a strongly connected component of $\mathcal{N}$ with no incoming edges has no outgoing edges: if it had, these would be unbounded, since initial liveness entails liveness for all strongly connected marked graphs. The marked graph produced by the transformation may thus be covered by strongly connected components, including isolated transitions as a particular case. Problem 2.3 may therefore be reduced to marked graphs covered by directed circuits, where at least one place is marked on each circuit (since all transitions are initially live), plus isolated transitions. In case when a marked graph satisfies these conditions but it is not connected, this marked graph may be seen as the *direct sum* $\mathcal{N} = \mathcal{N}_1 + \mathcal{N}_2$ of two smaller marked graphs $\mathcal{N}_1 = (P_1, T_1, F_1, m_{01})$ and $\mathcal{N}_2 = (P_2, T_2, F_2, m_{02})$, meaning that $\{P_1, P_2\}$ and $\{T_1, T_2\}$ are partitions of P and T, respectively, $F(p,t) = 0$ and $F(t,p) = 0$ whenever $p \in P_i$ and $t \in T_j$ with $i \neq j$, and the maps F_i and m_{0i} are the induced restrictions of F and m_0 on

$(P_i \times T_i) \cup (T_i \times P_i)$ and on P_i, respectively. The language $L(\mathcal{N})$ is then obviously the shuffle of the languages $L(\mathcal{N}_1)$ and $L(\mathcal{N}_2)$. Thus, if these languages are respectively equal to $L(\mathcal{N}'_1, \lambda_1)$ and $L(\mathcal{N}'_2, \lambda_2)$ for (disjoint) 1-safe marked graphs $\mathcal{N}'_1$ and $\mathcal{N}'_2$ and for labelling maps λ_1 and λ_2, we necessarily get $L(\mathcal{N}) = L(\mathcal{N}'_1 + \mathcal{N}'_2, \lambda_1 + \lambda_2)$ where $\lambda_1 + \lambda_2$ is the direct sum of the maps λ_1 and λ_2, *i.e.* $(\lambda_1 + \lambda_2)(t) = \lambda_i(t)$ for the unique $i \in \{1, 2\}$ such that $\lambda_i(t)$ is defined. The solution of problem 2.3 for a marked graph with no places and one (isolated) transition is trivial, any connected marked graph that may be covered by circuits is strongly connected, and any strongly connected marked graph is live if and only if it has no dead transitions. Therefore, it should be clear that problem 2.3 reduces to the following.

Problem 3.1 *Given any finite and bounded, live and strongly connected marked graph* $\mathcal{N} = (P, T, F, m_0)$*, is there a finite and 1-safe marked graph* $\mathcal{N}' = (P', T', F', m'_0)$ *and a transition labelling map* $\lambda : T' \to T$ *such that the labelled language* $L(\mathcal{N}', \lambda)$ *is equal to the free language* $L(\mathcal{N})$ *?*

4 From marked graphs to communicating sequential processes

A crucial fact that will considerably help us in solving problem 3.1 is the following: any strongly connected marked graph may be transformed into a language equivalent marked graph that can be covered by directed circuits *with disjoint sets of edges.* This fact does not need an elaborated proof, since it suffices to supplement each place with a complementary place if not already present, and then to decompose the marked graph into circuits of length 2, defined in an obvious way from pairs of complementary places. However, less systematic transformations to the same effect may be preferred, and we shall make no particular assumption on the length of circuits.

From now on, let $\mathcal{N} = (P, T, F, m_0)$ be the amalgamated sum on common transitions of a finite family of live and cyclic marked graphs $\mathcal{N}_i = (P_i, T_i, F_i, m_{0i})$

$$\mathcal{N} = \coprod \{\mathcal{N}_i \mid 1 \leq i \leq n\}$$

i.e. $\{P_1, \ldots, P_n\}$ is a partition of the set P, $T = T_1 \cup \ldots \cup T_n$, and the maps F and m_0 are the direct sums of the respective families of maps F_i and m_{0i}. Of course, a cyclic marked graph means here a marked graph which is at the same time a directed circuit. For each cyclic marked graph $\mathcal{N}_i$, let $|m_{0i}|$ denote the whole number of tokens in the subset of places P_i. Finally let M be the least common multiple of the numbers $|m_{0i}|$ for i ranging over $\{1, \ldots, n\}$.

Below in this section, we show that $\mathcal{N}$ may be simulated by a number of $\sum_i |m_{0i}|$ communicating sequential processes, each of which travels through a loop with length $(M/|m_{0i}|) \times |P_i|$ for the corresponding i. As a result, the overall size of the implementation is $M \times |P|$. It is worthwile trying to decrease M to the lowest possible value without changing $L(\mathcal{N})$. We shall not enter here into details, but it may be found suitable to modify for that purpose the initial marking m_0 of $\mathcal{N}$, by adding or removing tokens to or from any place p such that $m(p) \neq 0$ at every reachable marking m of $\mathcal{N}$.

Let us start looking for those communicating sequential processes. For notational convenience, we allow ourselves in the sequel to identify each place p in a marked graph with the unique pair of transitions (t, t') such that $F(t, p) = 1$ and $F(p, t') = 1$. The constructions we are about to explain rely on a (second) crucial observation, which is that $L(\mathcal{N})$ does not change when $\mathcal{N}$ and all marked graphs $\mathcal{N}_i$ are interpreted alternatively as Fifo-nets [4]. This amounts to map all tokens in m_0 injectively in $[1, |m_0|\,]$ and then consider each place $p = (t, t')$ as a fifo buffer, with transition t at the input and with transition t' at the output, such that *each token initially fed into place p stays forever in the (cyclic) marked graph* $\mathcal{N}_i$ *which p belongs to.* Thus, a transition t can be fired if and only if all its input buffers are non-empty, and firing this transition causes one token to flow from the input buffer of t to the output buffer of t in each marked graph $\mathcal{N}_i$ such that $t \in T_i$. At initialization, tokens are fed into buffers in arbitrary order. Nevertheless, this order is preserved to some extent under all firings. Actually, for each $i \in [1, n]$, let $\sigma_i = t_{i,1} \ldots t_{i,m_i}$ be a repetition free enumeration of T_i where each consecutive pair $p_{i,j} = (t_{i,j-1}, t_{i,j})$ is a place of $\mathcal{N}_i$ – thus $p_{i,1} = (t_{i,m_i}, t_{i,1})$ is also a place of $\mathcal{N}_i$. Then, the concatenation of all buffers $p_{i,j}$ from $j = 1$ to $j = m_i$, in this order, yields a sequence of tokens w_i invariant under all firings up to conjugation of words – where two words uv and vu are said to be *conjugates.*

Consider a fixed transition $t \in T$. Let $\{p_{i_1,j_1}, \ldots, p_{i_k,j_k}\}$ be the set of all buffers with transition t at the output. For each $l \in [1, k]$, let v_l be the sequence of tokens stored at initialization in the concatenated buffers of the fifo-net $\mathcal{N}_{i_l}$, starting this concatenation with the output buffer of transition t such that any two successive buffers are neighbours in this net. Let h_l be the last number in the sequence v_l. Thus, at the first time when transition t is fired, it will consume token h_l from buffer p_{i_l,j_l} for each $l \in [1, k]$. After this, the new sequence of tokens stored in the concatenated buffers of the fifo-net $\mathcal{N}_{i_l}$, starting again from the output buffer of transition t, is the conjugate of v_l obtained by shifting h_l from the last position to the first position. Therefore, one can make an exact list of all combinations of k tokens (from the respective fifo nets $\mathcal{N}_{i_l}$) that will actually enable the successive firings of transition t. This (cyclic) list starts with the k-tuple $(h_1, \ldots, h_k)$, read from the last positions of the respective sequences v_1 to v_k, and each subsequent item is obtained similarly after a simultaneous shift of all sequences by one position. As a result, the number K_t of items in this list is the least common multiple of the numbers $|m_{0i_l}|$ when l ranges over $[1, k]$. Denote by A_t the $(M \times k)$ array of token numbers obtained by repeating this list M/K_t times in sequence. Thus, each token owned by some given net $\mathcal{N}_i$ such that $t \in T_i$ occurs $M/\,|m_{0i}|$ times in A_t (in column l such that $i = i_l$).

We now convert arrays A_t to synchronization tables ST_t over new alphabets that will then be used to construct the sequential processes subject to synchronization (see the detailed example given in section 8). Each table ST_t is derived from the corresponding array A_t by replacing the successive occurrences of each token x, in column l where x appears, with new symbols $t(x, 1), t(x, 2), \ldots, t(x, M/\,|m_{0i}|)$, where $i = i_l$, in this order. The sequential processes are then defined as follows. Each fifo net $\mathcal{N}_i$ gives rise to as many processes $Pr(x)$

as there are tokens x in m_{0i}. For each token x stored at initialization in some buffer $p_{i,j}$ (with transition $t_{i,j}$ at the output), $Pr(x)$ is the cyclic process that iterates endlessly the finite sequence of actions as follows, where $L = M/|m_{0i}|$ and second subscripts are defined modulo the number m_i of transitions of $\mathcal{N}_i$:

$$\begin{array}{llllllllll} \underline{\text{do}} & t_{i,j}(x,1) & ; & t_{i,j+1}(x,1) & ; & \dots & ; & t_{i,j-1}(x,1) & ; & \\ & t_{i,j}(x,2) & ; & t_{i,j+1}(x,2) & ; & \dots & ; & t_{i,j-1}(x,2) & ; & \\ & \dots\dots\dots & . & \dots\dots\dots\dots & . & \dots & . & \dots\dots\dots\dots & . & \\ & t_{i,j}(x,L) & ; & t_{i,j+1}(x,L) & ; & \dots & ; & t_{i,j-1}(x,L) & & \underline{\text{od}} \end{array}$$

The collection of all processes $Pr(x)$ ($x \in [1, M]$), composed in parallel and synchronized according to the rows of the synchronization tables ST_t ($t \in T$), behaves indeed like the Petri net $\mathcal{N}$ as we show in the next section.

5 Establishing the simulation

We show first that every firing sequence of the *fifo* net $\mathcal{N}$ may be simulated by a run of the synchronized processes $Pr(x)$ producing the same trace (after replacing with t all the occurrences of synchronized actions from each table ST_t). We show next that every run of the synchronized processes $Pr(x)$ may be simulated by a firing sequence of the *Petri* net $\mathcal{N}$ with the same trace. As $L(\mathcal{N})$ is identical whether $\mathcal{N}$ is seen as a fifo net or not, an obvious conclusion follows.

Now for the first simulation. As a preliminary remark, observe that a given action $t(x,k)$ occurs exactly once in the cyclic process $Pr(x)$ and exactly once in the synchronization table ST_t. Each action $t(x,k)$ with token x owned by the net $\mathcal{N}_i$ conveys precisely the information that the token x is flowing for the k^{th} time through the transition t, where k is counted modulo $M/|m_{0i}|$. Each process $Pr(x)$ describes in this way the travel of the token x through the transitions of the net $\mathcal{N}_i$ in any run of the fifo net $\mathcal{N}$. Each table ST_t, when viewed as an implicit sequential and cyclic process, encodes similarly the projection of any run of the fifo net $\mathcal{N}$ on transition t. Each firing sequence of the fifo net $\mathcal{N}$ therefore induces by projection a synchronized run of all processes $Pr(x)$ with an identical trace.

If we had introduced one more sequential process per table ST_t, let

$$Pr(t) = \underline{\text{do}}\ ST_t[1]; ST_t[2]; \dots ; ST_t[M]\ \underline{\text{od}}\ ,$$

and synchronized every action $ST_t[l]$ from each process $Pr(t)$ with all actions $t(x,k)$ from the processes $Pr(x)$ such that $ST_t[l] = (\dots, t(x,k), \dots)$, a converse simulation of the system of all processes $Pr(x)$ and $Pr(t)$ by the fifo net $\mathcal{N}$ would immediately follow: sequences of actions of the synchronized processes would read directly as firing sequences of the fifo net $\mathcal{N}$. However, we prefer here interpreting tables ST_t as *sets* of synchronization vectors, defining the joint actions of the processes $Pr(x)$ without imposing *a priori* any sequential ordering between the joint actions defined in each table ST_t. The system under consideration is thus precisely the Arnold-Nivat product of all processes $Pr(x)$ with the rows of all tables ST_t

as synchronization vectors [1] [2]. Now, every sequence of synchronized actions performed from the initial state of this system is clearly mapped to a firing sequence of the Petri net $\mathcal{N}$ when each synchronized action $(\dots, t(x,k), \dots)$ is interpreted as firing t. Whenever a token x belongs to a subnet $\mathcal{N}_i$ and $t = t_{i,j}$, the readiness of the process $Pr(x)$ for performing $t(x,k)$ indicates that the token x is available from place $p = p_{i,j}$ at the input of transition t in the Petri net $\mathcal{N}$. The readiness of $Pr(x)$ for performing next $t_{i,j+1}(x,k)$ or $t_{i,j+1}(x,k+1)$ indicates that the token x is subsequently moved to place $p' = p_{i,j+1}$ (where the second subscript is counted modulo $m_i = |T_i|$).

Thus, if we let λ denote the labelling map that sends to t all row vectors in each table ST_t, then $L(\mathcal{N}) = \lambda(\mathcal{L})$ where $\mathcal{L}$ is the set of sequences of synchronized actions of the communicating sequential processes $Pr(x)$.

6 From the communicating sequential processes to a 1-safe labelled marked graph

It is pretty easy to tie the loose ends now by converting the synchronized product of processes $Pr(x)$ to a 1-safe marked graph $\mathcal{N}' = (P', T', F', m_0')$ with transition labelling map λ such that $L(\mathcal{N}', \lambda) = L(\mathcal{N})$, hence bringing a positive solution to problem 3.1.

Let $P' = \bigcup \{P_x' \mid x \in [1, |m_0|]\}$ where, for each process $Pr(x)$ with program $\underline{\text{do}}\quad t_{i,j}(x,1)\,;\, t_{i,j+1}(x,1)\,;\dots;\, t_{i,j-1}(x,1)\,;\dots;\, t_{i,j}(x,L)\,;\dots;\, t_{i,j-1}(x,L)\quad \underline{\text{od}}$, $P_x' = \{\, p_{i,j}(x,1)\,,\, p_{i,j+1}(x,1)\,,\dots,\, p_{i,j-1}(x,1)\,,\dots,\, p_{i,j}(x,L)\,,\dots,\, p_{i,j-1}(x,L)\,\}$ with $L = M/\,|m_{0i}|$. The initial marking of $\mathcal{N}'$ is defined on each subset P_x' by letting $m_0'(p_{i,j}(x,1)) = 1$ and leaving all other places $p_{i,l}(x,k)$ empty.

Let $T' = \bigcup \{T_t' \mid t \in T\}$ where each subset T_t' is the set of row vectors of the corresponding table ST_t. Let the labelling map λ be defined like in section 5, thus λ sends all items in each subset T_t' uniformly to the transition t of $\mathcal{N}$.

Finally, for each transition $t' = (\dots, t(x,k), \dots)$ in T', and for each entry $t(x,k)$ of this synchronization vector, let $F'(p,t') = 1$ and $F'(t',p') = 1$ where $p = p_{i,j}(x,k)$ if token x belongs to $\mathcal{N}_i$ and $t = t_{i,j}$, and p' is the successor of p in the enumeration of P_x'. Let $F'(p',t') = 0$ and $F'(t',p') = 0$ in all remaining cases.

Clearly, $\mathcal{N}'$ is a marked graph and this marked graph is 1-safe, since the initial marking of $\mathcal{N}'$ contains only one token per directed circuit. Also clearly, $L(\mathcal{N}') = \mathcal{L}$, and therefore $L(\mathcal{N}', \lambda) = \lambda(\mathcal{L}) = L(\mathcal{N})$. We now argue that the marked graph $\mathcal{N}'$ is *live*. In order to see this, suppose for a moment one adds to $\mathcal{N}'$ as many places as there are transitions in T', imposing on each subset of transitions T_t' an execution in the cyclic order defined by table ST_t, starting with the first item in this table. The augmented 1-safe marked graph $\mathcal{N}'$ behaves now exactly like the fifo net $\mathcal{N}$. As $\mathcal{N}$ is live, the augmented marked graph $\mathcal{N}'$ has at least one token on each directed circuit [5]. This property holds *a fortiori* for $\mathcal{N}'$, hence this marked graph is live.

7 Symmetries in the 1-safe marked graph

Since $\mathcal{N}'$ is a live and one-safe marked graph, $\mathcal{N}' = \sum_\gamma \mathcal{N}'_\gamma$ is a direct sum of strongly connected components $\mathcal{N}'_\gamma$ (hence, they also are live and one-safe marked graphs). We will show in this section that all components $\mathcal{N}'_\gamma$, considered as labelled marked graphs labelled with λ, are isomorphic up to initial markings. This fact, illustrated in the example given in section 8, depends on the working assumption that $\mathcal{N}'$ has been derived from a strongly connected marked graph $\mathcal{N}$, as was indeed assumed in the statement of problem 3.1.

To begin with, we make two easy remarks. First, each subset of places P'_x forms a directed circuit in the marked graph $\mathcal{N}'$. Therefore, for all $x \in [1, |m_0|]$, all transitions $t' \in T'$ such that $t' = (\ldots, t(x,k), \ldots)$ for some t and k are in the same component $\mathcal{N}'_\gamma$. Second, the free language $L(\mathcal{N})$ is the shuffle of all labelled languages $L(\mathcal{N}'_\gamma, \lambda)$, entailing that $L(\mathcal{N}'_\gamma, \lambda) \subseteq L(\mathcal{N})$ for each of them. Seeing that $\mathcal{N}$ is live and bounded, the maximal firing deviation between any two transitions of this net is bounded [5]. Therefore, by definition of the map λ, each component $\mathcal{N}'_\gamma$ has at least one transition from each subset T'_t.

Owing to the first remark, one may easily compute the set of transitions of each strongly connected component $\mathcal{N}'_\gamma$. For this purpose, let $\equiv$ be the least equivalence on the set $[1, |m_0|]$ of all token numbers such that $x \equiv y$ whenever x and y occur simultaneously in some row of some table ST_t. The equivalence classes of tokens are in bijection with the strongly connected components of $\mathcal{N}'$. Namely, for each class X of equivalent tokens, let $T'(X)$ ($\subseteq T'$) be the subset of all transitions t' in $T' = \bigcup \{T'_t \,|\, t \in T\}$ such that $t' = (\ldots, t(x,k), \ldots)$ for some $x \in X$ (with arbitrary t and k). Then $T'(X)$ is the set of transitions of some strongly connected component $\mathcal{N}'_\gamma$. The converse relation is obvious.

In order to show that all strongly connected components $\mathcal{N}'_\gamma$ are isomorphic up to initial markings, we intend to prove the following: if two row vectors $ST_t[j]$ and $ST_t[(j+l) \, mod \, M]$ from some table ST_t define two transitions in respective components $\mathcal{N}'_\alpha$ and $\mathcal{N}'_\beta$, then for every transition $ST_{t'}[j']$ of $\mathcal{N}'_\alpha$, $ST_{t'}[(j'+l) \, mod \, M]$ is a transition of $\mathcal{N}'_\beta$.

For simplicity, it is henceforth assumed that tokens have been numbered and distributed at the initialization of $\mathcal{N}$ into the various buffers $p_{i,j}$ such that each sequence of tokens w_i stored in $p_{i,1} \ldots p_{i,m_i}$ is the decreasing sequence $w_i = (s_i + |m_{0i}|) \ldots (s_i + 1)$, where $s_1 = 0$ and $s_{i+1} = |m_{01}| + \ldots + |m_{0i}|$ for positive i. In this way, each one of the arrays A_t considered in section 4 has successive rows

$$
\begin{array}{ccccc}
(f_1)^0(x_1) & \ldots & (f_l)^0(x_l) & \ldots & (f_k)^0(x_k) \\
(f_1)^1(x_1) & \ldots & (f_l)^1(x_l) & \ldots & (f_k)^1(x_k) \\
(f_1)^2(x_1) & \ldots & (f_l)^2(x_l) & \ldots & (f_k)^2(x_k) \\
\ldots\ldots\ldots & \ldots & \ldots\ldots\ldots & \ldots & \ldots\ldots\ldots \\
(f_1)^n(x_1) & \ldots & (f_l)^n(x_l) & \ldots & (f_k)^n(x_k) \\
\ldots\ldots\ldots & \ldots & \ldots\ldots\ldots & \ldots & \ldots\ldots\ldots
\end{array}
$$

where, for each token number x_l in between s_{i_l} (excluded) and s_{i_l+1} (included), $(f_l)^n(x_l)$ is the token number within the same interval such that $(f_l)^n(x_l)$ is congruent with x_l modulo $|m_{0i_l}|$.

Now let $ST_t[j] = (\ldots, t(x,k), \ldots)$ and $ST_t[(j+l)] = (\ldots, t(x',k'), \ldots)$ be two vectors in some table ST_t, with respective entries $t(x,k)$ and $t(x',k')$ in some fixed position. Let $s_i < x \leq s_{i+1}$, thus both tokens x and x' belong to subnet $\mathcal{N}_i$ of $\mathcal{N}$. It follows from the construction of synchronization tables that $l = (k'-k) \times |m_{0i}| + (x'-x)$. Hence, $x' = s_i + ((x - s_i + l) \, mod \, |m_{0i}|)$ and $k' = (k + \lfloor (x - s_i + l) \, / \, |m_{0i}| \rfloor)$ modulo $(M / \, |m_{0i}|)$. Observe that the computation of x' and k' does not depend on the chosen transition t of $\mathcal{N}$. Therefore, for each $l \in [1, M]$ and for all $t \in T$, vectors $ST_t[j]$ translate uniformly to vectors $ST_t[(j+l) \, mod \, M]$ by mapping each entry $t(x,k)$ with $s_i < x \leq s_{i+1}$ to $t(x',k')$ as described above. Moreover, whenever $t(x,k)$ has an immediate successor $t'(x,k)$ or $t'(x,(k+1) \, mod \, (M/ \, |m_{0i}|)$ in $Pr(x)$, $t(x',k')$ has the immediate successor $t'(x',k')$ or $t'(x',(k'+1) \, mod \, (M/ \, |m_{0i}|)$ in $Pr(x')$. Hence, every edge in $\mathcal{N}'$ with respective source and target $ST_t[j]$ and $ST_{t'}[j']$ is mapped in this way to an edge from $ST_t[(j+l) \, mod \, M]$ to $ST_{t'}[(j'+l) \, mod \, M]$. As each component $\mathcal{N}'_\gamma$ has at least one transition from each table ST_t, it follows obviously that all components $\mathcal{N}'_\gamma$ are isomorphic.

8 A detailed example

Consider the live and bounded marked graph $\mathcal{N}$ shown in Fig. 2. This strongly connected

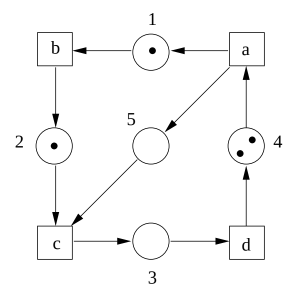

Figure 2: A live and bounded marked graph

marked graph cannot be covered by directed circuits with disjoint sets of edges. It suffices however to add one place, complementary to place 5, to obtain an equivalent marked graph that can be covered by two edge disjoint circuits as shown in Fig. 3. Denote α and β the tokens stored in places 1 and 2, respectively. Denote $\gamma;\delta$ and $\varepsilon;\zeta$ the sequences of tokens stored in places 4 and 6, respectively (place 6 is the new place complementary to place 5). Now $|m_{01}| = 4$ and $|m_{02}| = 2$ have the least common multiple $M = 4$. Therefore, the cyclic processes $Pr(x)$ induced from the six tokens $x \in \{\alpha, \beta, \gamma, \delta, \varepsilon, \zeta\}$ have programs as follows:

$Pr(\alpha) = \underline{\text{do}} \quad b\,(\alpha,1)\,;\, c\,(\alpha,1)\,;\, d\,(\alpha,1)\,;\, a\,(\alpha,1) \quad \underline{\text{od}}$
$Pr(\beta) = \underline{\text{do}} \quad c\,(\beta,1)\,;\, d\,(\beta,1)\,;\, a\,(\beta,1)\,;\, b\,(\beta,1) \quad \underline{\text{od}}$
$Pr(\gamma) = \underline{\text{do}} \quad a\,(\gamma,1)\,;\, b\,(\gamma,1)\,;\, c\,(\gamma,1)\,;\, d\,(\gamma,1) \quad \underline{\text{od}}$
$Pr(\delta) = \underline{\text{do}} \quad a\,(\delta,1)\,;\, b\,(\delta,1)\,;\, c\,(\delta,1)\,;\, d\,(\delta,1) \quad \underline{\text{od}}$

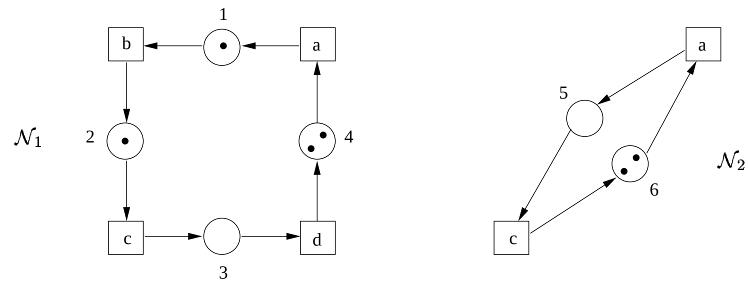

Figure 3: Two edge disjoint circuits

$Pr(\varepsilon) = \underline{\text{do}} \quad a\,(\varepsilon,1)\,;\, c\,(\varepsilon,1)\,;\, a\,(\varepsilon,2)\,;\, c\,(\varepsilon,2) \;\; \underline{\text{od}}$
$Pr(\zeta) = \underline{\text{do}} \quad a\,(\zeta,1)\,;\, c\,(\zeta,1)\,;\, a\,(\zeta,2)\,;\, c\,(\zeta,2) \;\; \underline{\text{od}}$

The synchronization tables ST_t from section 4 are computed as follows.

For $t = a$, $v_1 = \alpha\beta\gamma\delta$ and $v_2 = \varepsilon\zeta$, therefore $ST_a = \begin{pmatrix} a\,(\delta,1) & a\,(\zeta,1) \\ a\,(\gamma,1) & a\,(\varepsilon,1) \\ a\,(\beta,1) & a\,(\zeta,2) \\ a\,(\alpha,1) & a\,(\varepsilon,2) \end{pmatrix}$

For $t = c$, $v_1 = \gamma\delta\alpha\beta$ and $v_2 = \varepsilon\zeta$, therefore $ST_c = \begin{pmatrix} c\,(\beta,1) & c\,(\zeta,1) \\ c\,(\alpha,1) & c\,(\varepsilon,1) \\ c\,(\delta,1) & c\,(\zeta,2) \\ c\,(\gamma,1) & c\,(\varepsilon,2) \end{pmatrix}$

Finally, $ST_b = \begin{pmatrix} b\,(\alpha,1) \\ b\,(\delta,1) \\ b\,(\gamma,1) \\ b\,(\beta,1) \end{pmatrix}$ and $ST_d = \begin{pmatrix} d\,(\beta,1) \\ d\,(\alpha,1) \\ d\,(\delta,1) \\ d\,(\gamma,1) \end{pmatrix}$

The 1-safe marked graph $\mathcal{N}'$ that is derived from the above data as described in section 6 is shown in Fig. 4 –with the obvious labelling of transitions. $\mathcal{N}'$ is in fact the direct sum of two independent marked graphs, and these are actually isomorphic up to initial markings.

9 Conclusion

The main implication of the results established above is an algebraic characterization of the languages of bounded marked graphs. Let us recall De Simone's definition of the *mixed product* of two languages given in [3]. Let A_1 and A_2 be two alphabets, and let X and Y be languages of A_1^* and A_2^*, respectively. Their mixed product is the set of all words w in

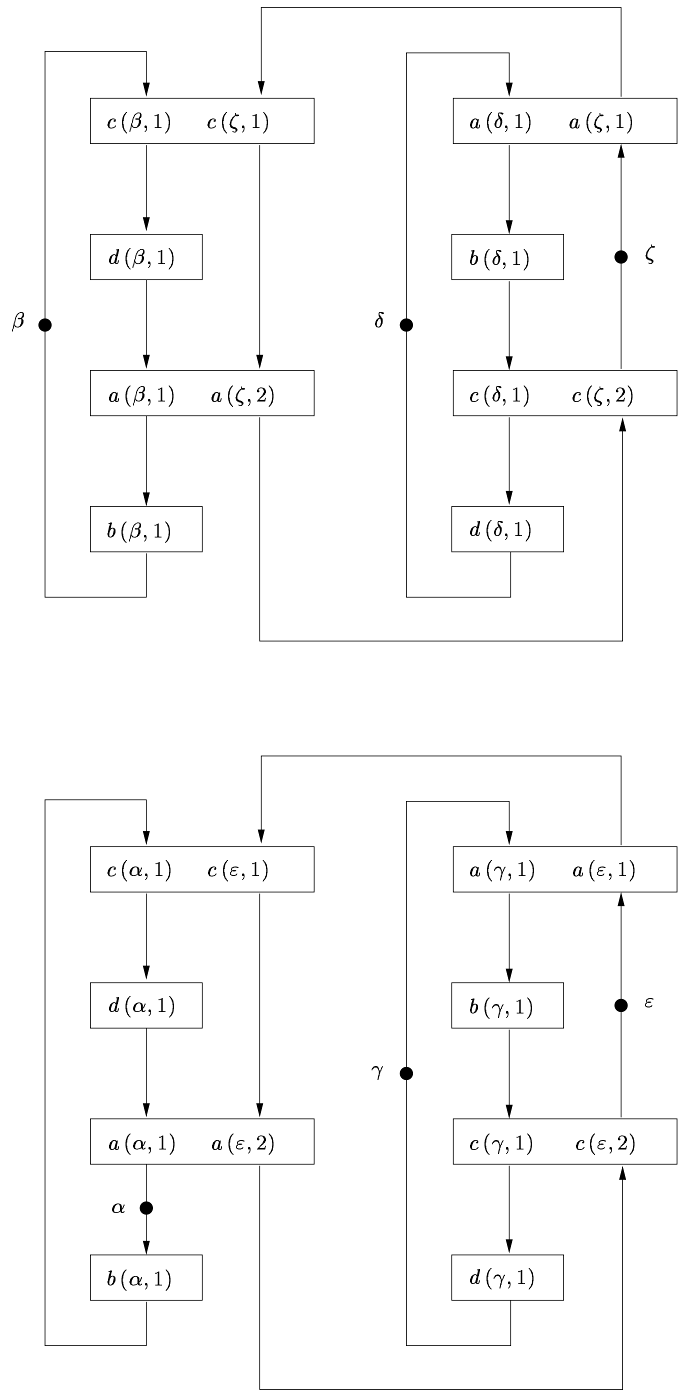

Figure 4: An equivalent 1-safe marked graph

$(A_1 \cup A_2)^*$ with respective projections $\pi_{A_1}(w) \in X$ and $\pi_{A_2}(w) \in Y$. Now, the languages of bounded marked graphs labelled in the alphabet Σ are the images under strict alphabetic morphisms $\phi : A^* \to \Sigma^*$ (where A is an arbitrary alphabet) of all languages of A^* defined from the set of generators $(a_1a_2)^*(\varepsilon + a_1)$, each one on some alphabet $\{a_1, a_2\} \subseteq A$, using mixed product as the composition operator.

References

[1] Arnold, A., Nivat, M.: Comportements de processus. Proc. Colloque AFCET "Les Mathématiques de l'Informatique" (1982) 35-68.

[2] Arnold, A.: Finite transition systems. Semantics of communicating systems. Prentice-Hall (1994).

[3] De Simone, R.: Langages infinitaires et produit de mixage. Theoretical Computer Science **31** (1984) 83-100.

[4] Finkel, A., Memmi, G.: Fifo Nets: A New Model of Parallel Computation. Proc. 6th GI-Conf. on Theor. Comp. Science, Springer-Verlag LNCS **145** (1982) 111-121.

[5] Murata, T.: Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE **77** no.4 (1989) 541-580.

[6] Reisig, W.: *Message posted on the PetriNets Mailing List on 22 Mar 2001*. Available from [http://www.daimi.au.dk/PetriNets/mailing-lists/archive.html] (2001).

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Unité de recherche INRIA Futurs : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399